



VIM Adaptation Layer for CORD

Canellas Cruz, Ferran; Kentis, Angelos Mimidis; Soler, José

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Canellas Cruz, F., Kentis, A. M., & Soler, J. (2018). *VIM Adaptation Layer for CORD*. Poster session presented at ONF Connect, Santa Clara, California, United States.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

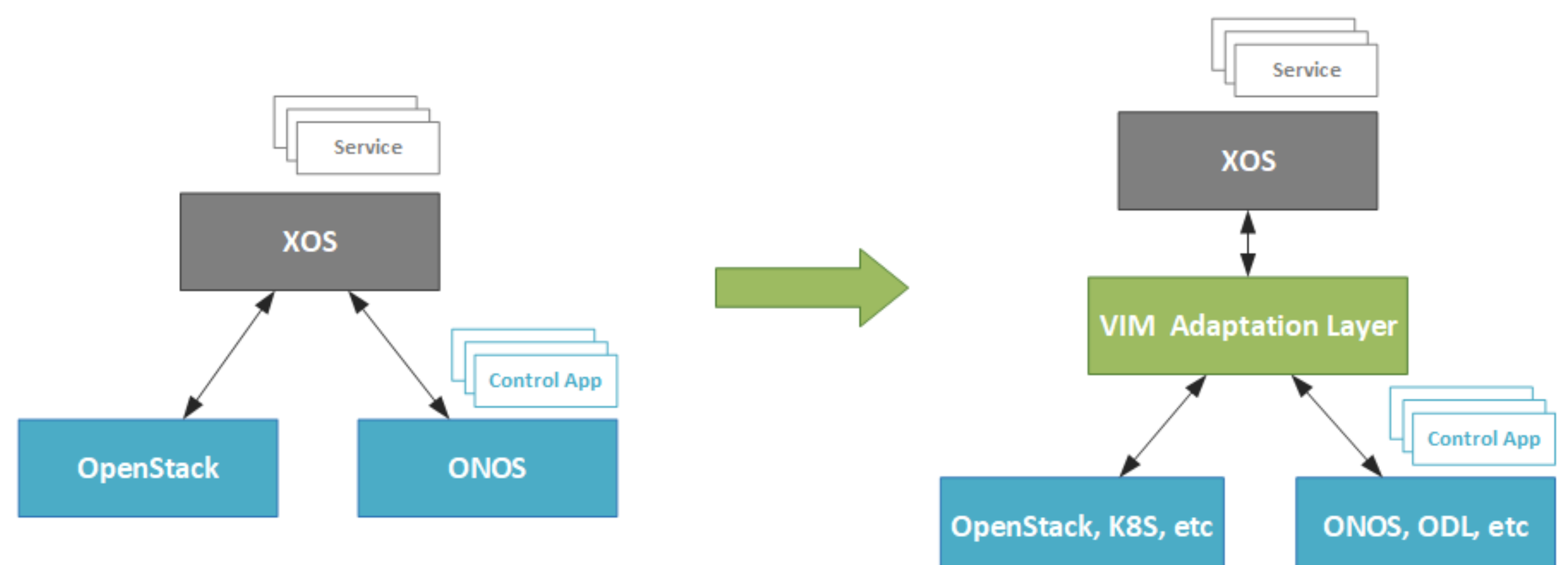
VIM Adaptation Layer for CORD

Problem Statement

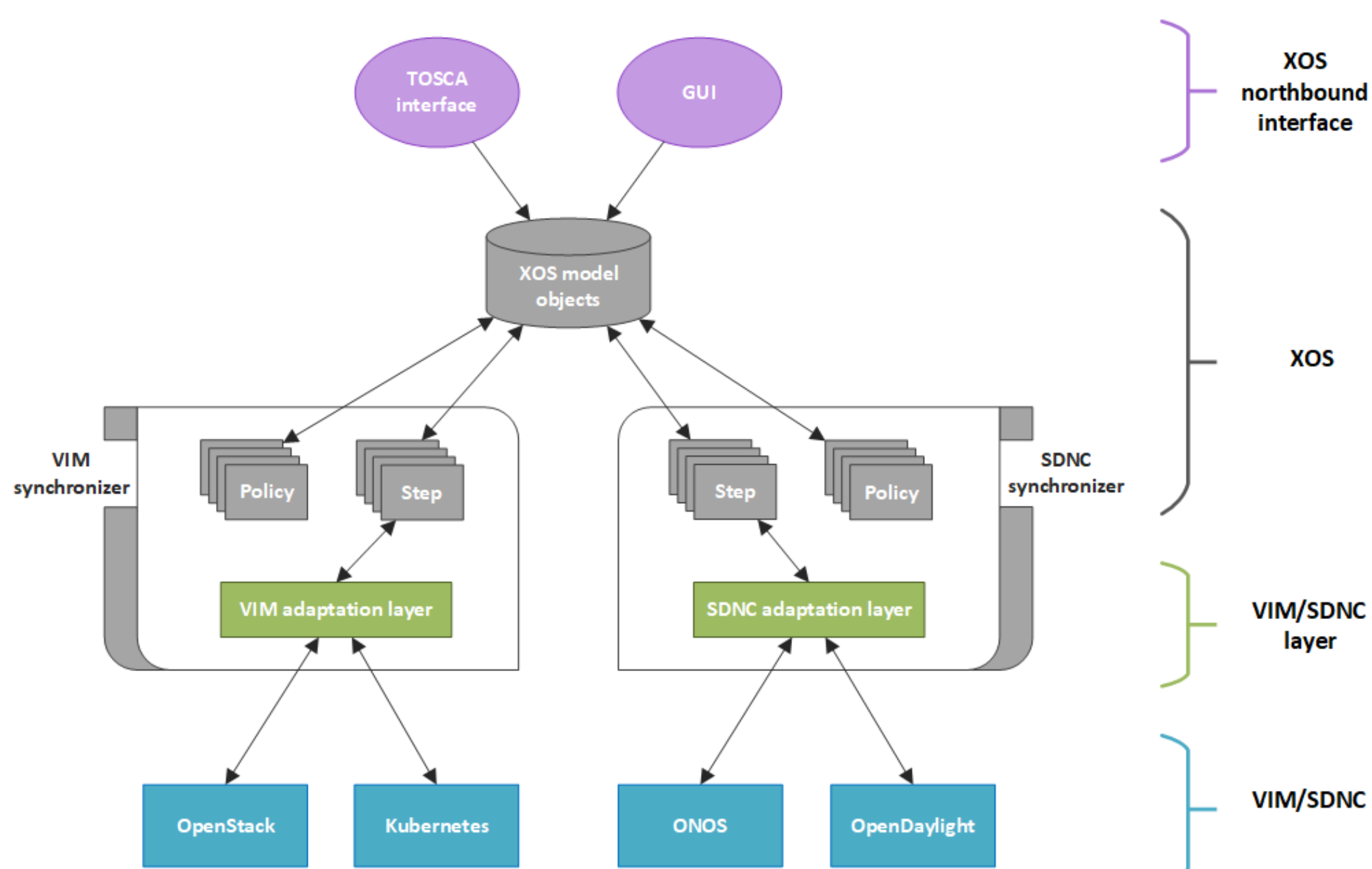
CORD 4.0 building blocks (XOS, OpenStack and ONOS) are tightly integrated and cannot be easily replaced or extended with alternative elements such as Kubernetes instead of OpenStack or OpenDaylight instead of ONOS. This penalizes CORD flexibility and therefore its adoption.

Objectives

1. Identify the interactions between XOS and OpenStack/ONOS by analyzing the internals of CORD 4.0.
2. Based on the identified interactions, define a generic interface that allows CORD to support multiple Virtual Infrastructure Managers, such as OpenStack, Kubernetes or Open Nebula.



VIM and SDNC Adaptation Layers



- Given the OpenStack and ONOS synchronizers, each step is grouped into a set of generic functions, that is, they are related to XOS models (create a Slice, get an Instance, delete a Port...) and take generic parameters.
- This set of generic functions conforms the VIM/SDNC layer interface.
- Instead, the synchronizer steps call the adaptation layer functions, which will run specific code depending on the underlying VIM/SDNC.
- Some XOS models should be updated to be generic enough.

VIM Layer Interface examples

```
create_slice(slice,controller)
add_user_role(role, user, controller,
site=None, slice=None)
get_networks(controller, network_id=None,
network_name=None)
delete_site(controller, site_id=None,
site_name=None)
```

SDNC Layer Interface examples

```
deploy_sdnc(sdnc_service)
add_configuration(sdnc_service, sdnc_app)
install_app(sdnc_service, sdnc_app)
activate_app(sdnc_service, sdnc_app)
```

Future work

- Adapt this work to CORD 6.0
- Implement the proposed adaptation layers

Project Info

Timeline	1.6.2017 – 31.05.2019
Website	http://ngpaas.eu
Twitter	http://twitter.com/NGPaaS_5GPPP
Youtube	https://www.youtube.com/channel/UCqIGeERucSED252rfUj8W6A
Contact	Ferran Cañellas: fccr@fotonik.dtu.dk Jose Soler : joss@fotonik.dtu.dk

This project is funded by the European Union's H2020-ICT-2016-2017 Programme under grand agreement no 761557